IN THE TITLE

Please replace the title with the following amended title:

LRU CACHE REPLACEMENT FOR A PARTITIONED SET ASSOCIATIVE CACHE

IN THE SPECIFICATION

Please replace the paragraph beginning on page 10, line 15, with the following amended paragraph:

The processor 30 comprises an in-order front end and an out-of-order back end. The in-order front end includes a bus interface unit 32, which functions as the conduit between the processor 30 and other components (e.g., main memory) of a computer system within which the processor 30 may be employed. To this end, the bus interface unit 32 couples the processor 30 to a processor bus (not shown) via which data and control information may be received at and propagated from the processor 30. The bus interface unit 32 includes Front Side Bus (FSB) logic 34 that controls communications over the processor bus. The bus interface unit 32 further includes a bus queue 36 that provides a buffering function with respect to communications over the processor bus. The bus interface unit 32 is shown to receive bus requests 38 from, and to send snoops or bus returns 40 to, a memory execution unit 42 that provides a local memory capability within the processor 30. The memory execution unit 42 includes a unified data and instruction cache 44, a data Translation Lookaside Buffer (TLB) 46, and memory ordering buffer 48. The memory execution unit 42 receives instruction fetch requests ~~50~~ from, and delivers raw instructions ~~52~~ (i.e., coded macroinstructions) to, a microinstruction translation engine 54 that translates the received macroinstructions into a corresponding set of microinstructions.

Please replace the paragraph beginning on page 11, line 12, with the following amended paragraph:

The microinstruction translation engine 54 effectively operates as a trace cache "miss handler" in that it operates to deliver microinstructions to a trace cache 62 in the event of a trace

cache miss. To this end, the microinstruction translation engine 54 functions to provide the fetch and decode pipe stages 12 and 14 in the event of a trace cache miss. The microinstruction translation engine 54 is shown to include a next instruction pointer (NIP) 100, an instruction Translation Lookaside Buffer (TLB) 102, a branch predictor 104, an instruction streaming buffer 106, an instruction pre-decoder 108, instruction steering logic 110, an instruction decoder 112, and a branch address calculator 114. The next instruction pointer 100, TLB 102, branch predictor 104 and instruction streaming buffer 106 together constitute a branch prediction unit (BPU) ~~99~~. The instruction decoder 112 and branch address calculator 114 together comprise an instruction translate (IX) unit ~~113~~.

Please replace the paragraph beginning on page 20, line 6, with the following amended paragraph:

As alluded to above, the trace delivery engine 60 may function as a primary source of microinstructions during periods of high performance by providing relatively low latency and high bandwidth. Specifically, for a CISC instruction set, such as the Intel Architecture x86 instruction set, decoding of macroinstructions to deliver microinstructions may introduce a performance bottleneck as the variable length of such instructions complicates parallel decoding operations. The trace delivery engine 60 attempts to address this problem to a certain extent by providing for the caching of microinstructions, thus obviating the need for microinstructions executed by the execution unit ~~17~~ to be continually decoded.